RISPOSTE DOMANDE APERTE CYBERSECURITY

1) **List and briefly define categories of security services.**
   Le categorie dei security services sono:
   1. Confidentiality: protezione dei dati contro la loro divulgazione non autorizzata.
   2. Data Integrity:
   3. Message Authentication: garantire che il mittente di un messaggio sia effettivamente chi dichiara di essere e che il contenuto del messaggio non sia stato alterato durante il transito.
   4. Peer entity authentication: verificare che l'entità con cui si sta comunicando sia effettivamente quella che dichiara di essere.
   5. Authorization: verifica dei permessi di accesso ad una risorsa.
   6. Access Control: abilità di limitare e controllare gli accessi ai sistemi.
   7. System Integrity: proteggere i sistemi da cambiamenti non autorizzati.
   8. Availability: capacità di accedere e utilizzare le risorse di un sistema informatico o di una rete quando necessario, senza interruzioni o ritardi significativi (ad esempio dovuti ad attacchi Denial Of Service).
   9. Accountability e Non Repudiation: garantiscono che le parti coinvolte in una transazione o una comunicazione siano responsabili delle loro azioni e non possano negare la loro partecipazione o le informazioni fornite.
   10. Anonymity: la condizione di un'identità di essere sconosciuta o nascosta (Un'applicazione potrebbe voler mantenere l'anonimato degli utenti o di altre entità del sistema, magari per preservarne la privacy)

2) **What is the difference between passive and active security threats?**
   Gli attacchi passivi sono tattiche che mirano a intercettare o monitorare le comunicazioni o le risorse senza apportare modifiche dirette, gli aggressori cercano di ottenere informazioni o accesso non autorizzato senza lasciare tracce evidenti (interception, traffic analysis, sniffing di rete).
   Gli attacchi attivi: implicano l'interazione diretta con i sistemi di destinazione al fine di alterare, danneggiare o compromettere la sicurezza delle risorse. (Spoofing, tampering, replay, Repudiation DoS)

3) **What is the difference between a block cipher and a stream cipher?**
   Gli stream cipher processano I messaggi un bit o un byte alla volta, l'unità di d'output C_i può essere funzione dell'input corrente M_i, di uno stato interno S_i e della chiave segreta K, che idealmente deve essere lunga quanto il messaggio M (OTP). Molti degli stream ciphers sono basati su PRNG (pseudo-random-number-generator) in cui la key inizializza il generatore e sia i byte della chiave che del testo sono usati per produrre flussi di byte.
   I block cipher processano i messaggi in blocchi, ognuno dei quali viene criptato/decriptato. Sia il plaintext che ciphertext sono trattati come sequenze di n-bit a blocchi di dati, il ciphertext è della stessa lunghezza del plaintext. I messaggi lunghi devono essere processati in blocchi (se la dimensione del messaggio non è divisibile per il numero dei blocchi può essere applicato del padding). Se n è la dimensione del cifrario e k la dimensione della chiave si avranno 2^k possibili trasformazioni.

4) **List and briefly define types of cryptanalytic attacks based on what is known to the attacker.**
   Gli attacchi cryptanalytic tentano di sfruttare alcune caratteristiche dell'algoritmo oppure alcune coppie di ciphertext o plaintext per dedurre la key o il plaintext. In base alle caratteristiche possedute da un attaccante abbiamo:
   1. Ciphertext only attack: l'attaccante ha solo del ciphertext a disposizione, dovrebbe essere in grado di riconoscere quando ha avuto successo (ad esempio conosce il formato del documento oppure riconoscimento di un testo). È l'attacco più difficile in quanto è richiesta una grande quantità di dati.
   2. Known Plaintext attack: l'attaccante conosce una coppia (plaintext, ciphertext), dalla coppia l'attaccante può mappare alcune parti del testo. Per ottenere il plaintext l'attaccante potrebbe già sapere cosa contiene il messaggio (parole chiavi, patterns noti etc.)
   3. Chosen Plaintext (o ciphertext) attack: l'attaccante può scegliere qualsiasi plaintext del messaggio ed ottenere il ciphertext corrispondete (o il contrario), ad esempio un attaccante che entra in un servizio di trasmissione del sistema e chiede di trasmettere qualsiasi plaintext.
5) **What is the difference between an unconditionally secure cipher and a computationally secure cipher?**
   Un unconditionally secure cipher è uno schema d'encryption in cui il cifrario non può subire un break (non importa quanta potenza computazionale si usa per l'attacco). Es. One Time Pad
   Un computationally secure cipher indica uno schema d'encryption che, se attaccato con un limitato numero di risorse computazionali, il cifrario non può subire un break.

6) **Briefly define the monoalphabetic cipher.**
   Un cifrario monoalphabetic applica delle permutazioni arbitrarie al testo in input. La sostituzione può essere vista come la chiave segreta (26! = 4x10^26 chiavi diverse)
   Il problema di questi tipo di cifrari è la ridondanza del testo dovuta dalla distribuzione non uniforme della frequenza dei caratteri -> vulnerabile ad attacchi criptoanalitici (prima si calcola la frequenza delle lettere per il ciphertext, poi si comparano i conteggi con i valori noti delle frequenze ed infine si creano delle tabelle di singole, doppie e triple frequenze)

7) **What is the avalanche effect for an encryption algorithm?**
   L'avalanche effect è una proprietà desiderabile delle key degli algoritmi d'encryption, per cui cambiare un input o una chiave di un bit porti al cambiamento della metà dei bits degli output, questa caratteristica caratterizza i block ciphers ma non gli stream cipher.

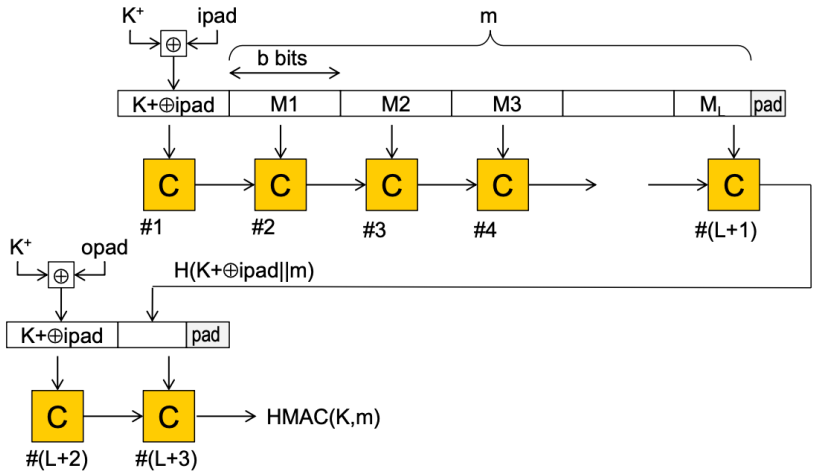
8) **Which of the following block cipher modes of operation ECB, CBC, OFB, CFB, CTR, use only the encryption function for both encryption and decryption?**
   l'Output Feed-back (OFB), il Cipher Feed-back (CFB) ed il Counter Mode (CTR) usano la funzione XOR sia per l'operazione di encryption che per la decryption sfruttando la proprietà dell'inversa dello XOR. Se C= A XOR B -> A = C XOR B e B = C XOR A

9) **What characteristics are needed in a secure hash function?**
Una hash function sicura dovrebbe rispettare le seguenti caratteristiche:
Preimage Resistance: Dato un output hash, dovrebbe essere computazionalmente impossibile determinare l'input originale che ha prodotto quel valore hash specifico.
Weak collision resistance: è computazionalmente irrealizzabile trovare un secondo input la cui hash function ha lo stesso risultato di un input specifico, ovvero dato un input m trovare m' diverso da m tale che H(m') = H(m).
Strong collision resistance: è computazionalmente impossibile trovare due input m, m' distinti che hanno l'hash che restituisce lo stesso output.

10) **List possible usages of hash function.**
I possibili utilizzi dell'hash function sono:
1. Message fingerprint: verificare l'integrità di un dato confrontando il message digest salvato in qualche dato/programma al posto del messaggio intero.
2. Password sharing: un sistema potrebbe conservare solo l'hash della password.
3. Firma digitale: usato per firmare solo il message digest al posto del messaggio intero, rendendo l'operazione più semplice.
4. Autentificazione d'identità:
5. Message authentication: H(m) può essere usato in un MIC per m, però se H(m) non viene protetto può essere modificato da un intruso.
6. Encryption: una funzione H può essere usata per costruire un cifrario (ad esempio un one time pad generato da uno pseudorandom bit-stream proveniente da un HASH del segreto e criptato con il messaggio con uno XOR con lo stream)

11) **What is a message authentication code?**
Il message authentication code (MAC) è una Keyed hash function, una funzione one way che crea un piccolo blocco di dimensione fissa che dipende da un messaggio d'input m ed una chiave k: F_K(m) = F(K,m).
Condensa un messaggio di lunghezza variabile m in un blocco di dimensione fissa, è usato come message authenticator. Il modo più semplice per generare un MAC è combinare un hash function con la propria secret key (HMAC).

12) **Consider a hash function H(.) that iteratively processes the input message in blocks of size M using a compression function C (an example could SHA1 that uses a C function of size 512 bits). Show the scheme of HMAC as function of H, and evaluate the number of C passes (number of calls of the function C) required to calculate the HMAC of a message m with length N.M.**
Se servono N passi per calcolare l'Hash, allora il calcolo dell'Hmac prende solo 3 passi in più-> N+3



13) What is the meaning of the expression a≡b (mod n)?

L'espressione indica che a è congruente modulo n a b: se a e b sono divisi per n il loro rapporto ha resto uguale. Inoltre (a-b) è divisibile per n, quindi esiste un intero k tale che (a-b) = kn.

14) How is it defined the multiplicative inverse of an integer modulo n?

L'inverso moltiplicativo di un numero x è il numero che deve essere moltiplicato a x per ottenere 1, ovvero u: u*m = 1 mod(n). È possibile usare l'algoritmo di Euclide esteso per calcolarlo, inoltre m ha un moltiplicativo inverso se e solo se m ed n sono coprimi.

15) What is Euler's totient function?

L'Euler's totient function phi(n) è il numero di elementi presenti nell'insieme dei residui in modulo n, ovvero l'insieme composto dai residui che sono primi relativi ad n (se il loro GCD con n è pari ad 1). Per calcolarlo bisogna contare il numero di elementi da escludere dall'insieme completo dei residui utilizzando la fattorizzazione prima. Phi(p,q) con p e q relativamente primi è pari a (p-1)(q-1).

16) What is the discrete logarithm?

Il logaritmo discreto x =log_a(b)mod(p) è una funzione matematica che consiste nel trovare x dove a^x = b(mod p). Se a è una primitive root e p è primo allora x esiste sempre; nel caso di p non primo, x esiste se a è una primitive root e b è co-primo con n. Trovare il logaritmo discreto è un problema difficile utilizzato spesso nell'ambito della cybersecurity (es. Diffie-Hellman key exchange).

17) Describe RSA encryption algorithm.

L'RSA è un algoritmo che fa parte della criptografia a chiave pubblica, o asimmetrica, in cui chi cripta i messaggi o verifica le firme non può decriptarli o creare le firme.
Si basa sull'elevamento a potenza in un campo finito (Galois) su numeri interi modulo n, ovvero O(log(n))^3) operazioni. Usa numeri interi di 1024 bit, che garantiscono la sicurezza dell'algoritmo (la fattorizzazione comporta O(e^(log(n)log(n)log(n)) operazioni). La lunghezza della chiave e del plaintext sono variabili, il plaintext deve essere meno lungo della chiave mentre il ciphertext ha lunghezza pari a quella della chiave. La generazione di una chiave pubblica e della corrispondente chiave privata avviene mediante le seguenti fasi:

1. Vengono scelti due numeri primi grandi p e q (512 bits ciascuno) che rimarranno segreti
2. I due numeri vengono moltiplicati ottenendo un numero n (1024 bits)
3. Viene calcolata l'Eulero totient function phi(n) = (p-1)(q-1)
4. Si sceglie un numero e relativamente primo a phi(n)
5. Si trova d che è l'inverso moltiplicativo di e mod(phi(n))

La chiave pubblica K_U = <e,n>
Mentre la chiave privata K_R = <d,n> (oppure <d,p,q>)
Per criptare il messaggio: c = m^e mod n
Per decriptare: m = c^d mod n
Il funzionamento dell'RSA è garantito dal teorema di Eulero, se diverse chiavi condividono lo stesso modulo n ed un avversario conosce c_1, c_2 (ciphertext diversi) e le loro chiavi pubbliche, può recuperare m (plaintext originale). Il valore di e non deve essere troppo piccolo altrimenti l'algoritmo è facilmente attaccabile, stessa cosa vale per m.

18) Describe Diffie-Hellman key exchange.

Diffie-Hellman è un metodo pratico per lo scambio pubblico di chiavi segrete (no encryption, no firma digitale). Permette a due individui di concordare un segreto condiviso, tutti gli utenti condividono parametri globali: p (numero o polinomio intero primo), e g primitive root mod p. Ogni utente genera la propria chiave x_A<p e calcola la sua chiave pubblica y_A = g^(x_A), che viene resa pubblica da ogni utente.

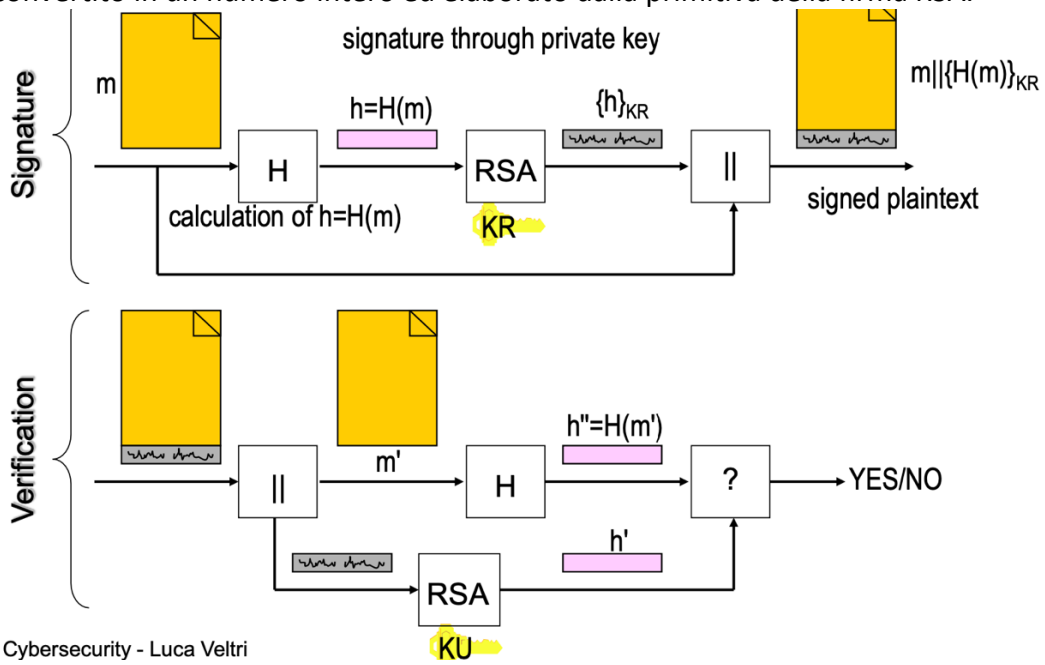
Le chiavi condivise K_AB = g^(x_A*x_B)mod(p) che è uguale a:

- K_AB = y_B^(x_A) -> calcolabile da A che conosce y_B e x_A
- K_AB = y_A^(x_B) -> calcolabile da B che conosce y_A e x_B

K_AB può essere usata come una session key in uno schema segreto d'encryption tra e B. Il metodo è vulnerabile all'attacco man in the middle, se il canale di comunicazione non è protetto correttamente.

19) Show how RSA digital signature and verification work.

L'RSA digital signature definisce due metodi di encoding per firme con appendice, viene applicata un'operazione di codifica del messaggio per produrne uno codificato, che viene convertito in un numero intero ed elaborato dalla primitiva della firma RSA.

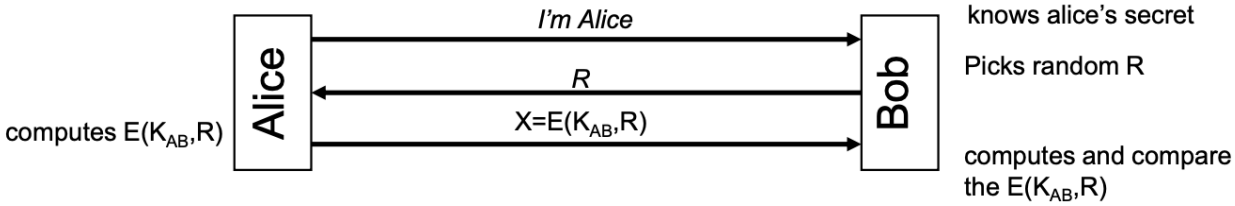


Cybersecurity - Luca Veltri

20) Show possible challenge-response identification schemes based on symmetric cipher, asymmetric cipher, hash function, MAC function, or digital signature.

L'identificazione challenge-response si basa sull'entità che si dichiara che prova la sua identità ad un'altra dimostrando la conoscenza del segreto senza rivelarlo al verificatore. Ciò viene fatto rispondendo ad una challenge che varia nel tempo per cui la risposta dipende sia dalla challenge stessa che dal segreto dell'entità.

Questo è un esempio challenge-response con chiave simmetrica in cui la funzione E() di encryption potrebbe essere sostituita con un Hash Function oppure un MAC.

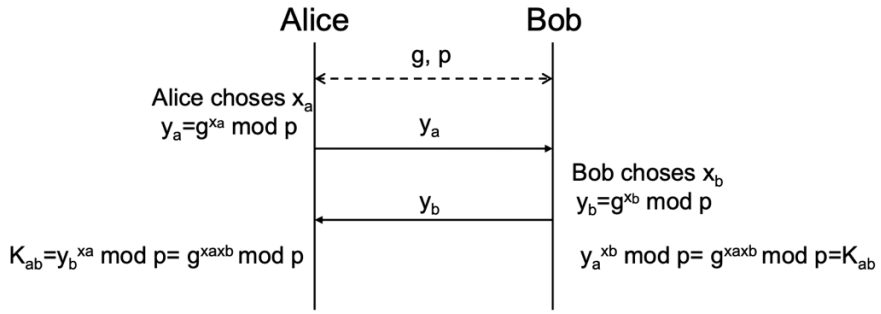


Con X= MAC(K_(AB),R) oppure X= H(secret_(AB) || R).
ESEMPIO per firma digitale: un server genera una challenge casuale e la invia al client. Il client firma la challenge utilizzando la sua chiave privata e invia la challenge insieme alla firma al server. Il server verifica la firma utilizzando la chiave pubblica del client e controlla se corrisponde alla sfida originale.

21) Show different ways in which secret keys can be established between two parties.

I metodi principali per lo scambio di chiavi sono:

- POINT-TO-POINT KEY TRANSPORT: basato su key simmetriche long term condivise "a priori" tra due parti A e B, distribuite inizialmente su un canale sicuro. Versione più semplice: one passage A -> B: E_K_(ab)(K_S) dove K_S prende il nome di session key. Un'altra variante potrebbe essere quella con il challenge-response in cui avvengono due messaggi A <- B: n_B e la risposta A -> B: E_K(ab)(K_S, n_B, B*)
- DYNAMIC KEY DERIVATION: la session key in questo caso si basa su input casuali per-sessione forniti da una delle due parti, A -> B: r_A, e K_S = f(r_A) con f una funzione crittografica come MAC() oppure E_K.
- DIFFIE-HELMANN:
  si basa sul seguente schema:

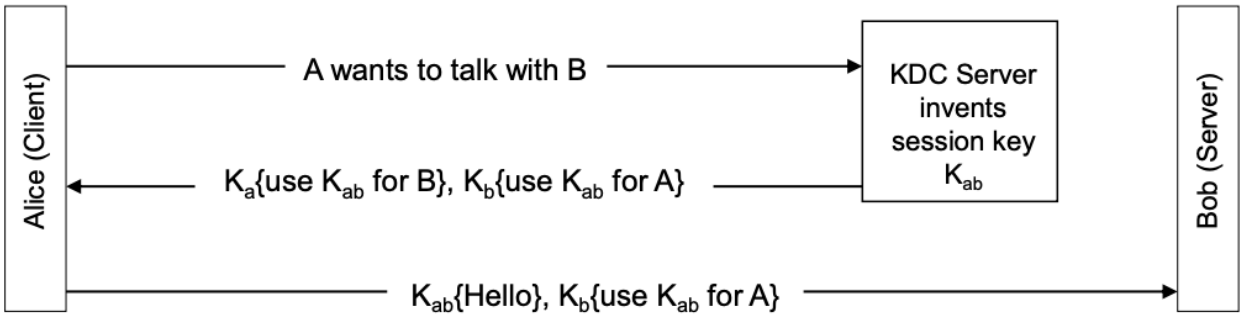


22) What is the difference between a short-term key (session key) and a long-term key?

Una short term key è usata per comunicazioni sicure, hanno un life-time limitato di tempo mentre le long term keys sono le chiavi usate per la peer authentication o per gli scambi di keys sicuri. Per il passaggio sicuro di LTK si usa la key pre-distribution mentre per le STK si usano dei metodi di Dynamic Key Establishment (Key Transport o Key Agreement)

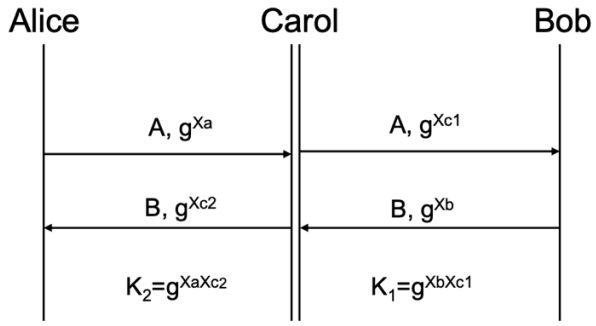
23) What is a key distribution center? Describe a possible key distribution scheme that uses a KDC.

Un centro di distribuzione delle chiavi (KDC) è un'autorità centrale o un server responsabile della distribuzione e della gestione sicura delle chiavi crittografiche in una rete. Agisce come una terza parte fidata, facilitando la comunicazione sicura tra più entità generando e distribuendo chiavi di sessione. Un possibile schema di utilizzo è il seguente (kerberos protocol):



24) Why Diffie-Hellman key exchange is vulnerable to Man-in-the-middle attack? Show an example of successful MITM attack to the DH.

Lo scambio di chiavi Diffie-Hellman è vulnerabile a un attacco Man-in-the-Middle (MITM) perché le chiavi pubbliche scambiate durante il protocollo non sono autenticate. Un utente malintenzionato può intercettare e modificare le chiavi pubbliche scambiate tra le due parti, inserendosi come entità "intermedia" e stabilendo chiavi segrete separate con entrambe le parti. Ecco un esempio:



25) Describe how the Diffie-Hellman key exchange can be generalized for group key exchange.

Esempio con 4 utenti:

According to GDH, all participants should agree on a prime number p and generator g. Each participant (u_i) generates a secret (private) value x_i.
The finally group key will be $g^{x1x2x3x4} \bmod p$. For computing such key, each participant should receive $y=g^{a\,b\,c} \bmod p$ with $a,b,c \neq x_i$ and use x_i for computing $y^{xi} \bmod p = g^{a\,b\,c\,xi} \bmod p = g^{x1x2x3x4} \bmod p$

One possible message exchange could be:
$u1 \rightarrow u2 : g^{x1}$
$u2 \rightarrow u3 : g^{x1x2}, g^{x1}$
$u3 \rightarrow u4 : g^{x1x2x3}, g^{x1x2}, g^{x1}$
$u4 \rightarrow u3 : g^{x1x2x4}, g^{x1x4}, g^{x4}$
$u3 \rightarrow u2 : g^{x1x3x4}, g^{x3x4}$
$u2 \rightarrow u1 : g^{x2x3x4}$

26) What is a digital certificate?

Un certificato digitale è un documento digitale che certifica l'associazione di una public key con il suo proprietario: cert_c(A) = {X={K_A,ID_A,ID_C},sign_K_C(X)}, può includere la chiave pubblica K_a, il nome del proprietario della chiave pubblica (ID_A, il nome dell'emittente del certificato ID_C, la firma digitale dell'emittente, ed anche altre possibili

informazioni (numero seriale del contratto, data emissione/scadenza etc.). L'emittente è una 3° parte fidata, i certificati digitali sono usati per la firma dei documenti, la distribuzione delle chiavi e l'autenticazione di entità.

27) What is a chain of certificates?

Se un'entità B volesse ottenere la chiave pubblica di A, generalmente ha bisogno di ottenere e validare il relativo certificato digitale; per farlo B deve conoscere la public key di C1 che ha emesso il certificato di A. Nel caso non conoscesse la public key di C1 basterebbe utilizzare il certificato di C1 (affidandosi ad un'altra entità C2 etc.), è possibile rappresentare la situazione attraverso una catena di certificati: cert_CN(CN-1), cert_(CN-1)(CN-2), …, cert_C2(C1), cert_C1(A) -> a B gli basta conoscere la Public Key della prima entità C_n della catena. (Altra possibile rappresentazione è mediante un grafo in cui i vertici sono le entità e gli archi i certificati digitali)

28) What information is included in a X.509 certificate?

X.509 è uno standard per una public key infrastructure (trust model in cui le chiavi pubbliche sono firmate solo da Certification Authorities). Un certificato digitale X.509 include:

- Public key e public key info
-Distinguished Name e altre informazioni sul proprietario del certificato.
-Distinguished Name e altre informazioni sul proprietario dell'emittente del certificato.
-Version number del certificato.
-Serial Number associato al certificato.
-Livello di fiducia
-Data emissione
-Data scadenza
-Firma digitale (incluse info sull'algoritmo utilizzato per la firma)
-altre estensioni

29) What is a X.509 Certification Revocation List (CRL)?

La CRL è una struttura dati emesse e firmata periodicamente da ogni CA. Si tratta di una lista time-stamped che identifica i certificati revocati e che è firmata da un CA e resa liberamente disponibile su una repository pubblica. Quando un sistema usa un certificato oltre a controllarne la firma, acquisisce l'ultima versione disponibile del CRL e controlla che non sia presente il serial number del certificato.
Un'aggiunta sul CRL viene messa come parte del prossimo aggiornamento dopo la notifica di revoca, una voce della lista potrebbe essere rimossa dopo essere apparsa su una CRL regolarmente programmata emessa oltre il periodo di validità del certificato revocato.
Una limitazione del metodo di revoca CRL è che la granularità temporale della revoca è limitata al periodo d'emissione della CRL.

30) Which are the security services provided by IPSec ESP?

IPSec, è un framework per fornire una sicurezza interoperabile, di alta qualità e basata sulla crittografia per IPv4 e IPv6. IPSec ESP è un protocollo che offre integrità, autenticazione dell'origine dei dati, riservatezza dei dati, un servizio opzionale anti-attacchi replay ed un traffic flow confidentiality service. Tutti i protocolli IPSec (incluso ESP) sono progettati per essere indipendenti dagli algoritmi crittografici utilizzati.

31) Which are the security services provided by TLS?

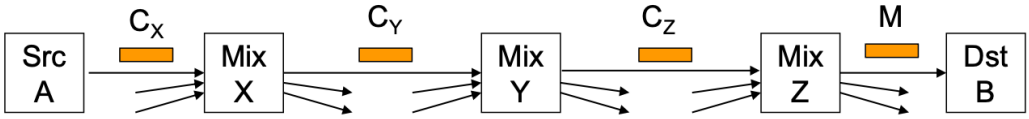
Il protocollo Transport Layer Security (TLS) garantisce la privacy e l'integrità dei dati tra due applicazioni client/server comunicanti. Nel dettaglio fornisce: autenticazione dell'identità dei peer (mediante algoritmi di crittografia asimmetrica come RSA, DSA, etc.) e negoziazione di una chiave segreta sicura (TLS Handshake Protocol), confidenzialità fornita dall'utilizzo di crittografia simmetrica (TLS Record Protocol), message integrity e data authentication perché il trasporto dei messaggi avviene utilizzando dei MAC (creati con funzioni hash).

32) Which certificate is usually involved in a TLS handshake between a client and a server?

Tutti i certificati in TLS sono nel formato X.509, per verificare che un certificato sia valido, il verificatore deve controllare che la firma della CA sia valida e verificare che il proprietario del certificato conosca la chiave privata. Verifica, inoltre, che le informazioni identificative siano quelle che dovrebbero essere. Alcuni certificati Root/CA sono preinstallati: Firefox, Opera e IE hanno elenchi diversi. La chiave pubblica della CA viene utilizzata per verificare le firme sui certificati emessi. I browser possono accettare certificati non verificabili o avvisare l'utente.

33) Describe a possible scheme for a high-latency anonymity system based on multiple mixes (mix network).

Un Mix è l'elemento costitutivo di base di quasi tutti gli high-latency anonymity systems. È un processo che accetta messaggi crittografati come input e raggruppa diversi messaggi insieme in un batch; quindi, decripta ed inoltra alcuni o tutti i messaggi nel batch. L'utilizzo di un singolo mix porta non solo a un singolo punto di errore, ma anche a un singolo punto di fiducia; per questo motivo vengono progettati i mix networks in modo che il mandante possa scegliere un insieme di mix a cui mandare un messaggio.
Se Alice vuole inviare anonimamente un messaggio M a Bob tramite un percorso P = {X, Y, Z}: Creerebbe iterativamente un livello di crittografia per ogni mix iniziando con l'ultimo del percorso e lavorando a ritroso verso il primo: EK_x(ID_Y,EK_y(ID_Z,EK_z(ID_B,M)).
Alice invia quindi il testo cifrato al primo mix nel percorso e, una volta ricevuto, ogni mix può rimuovere un livello di crittografia per estrarre l'indirizzo del mix successivo nel percorso e un messaggio di testo cifrato da inoltrare a quel mix.



- $C_X = E_{Kx}(ID_Y \,||\, C_Y) = E_{Kx}(ID_Y \,||\, E_{Ky}(ID_Z \,||\, E_{Kz}(ID_B \,||\, M)))$
- $C_Y = E_{Ky}(ID_Z \,||\, C_Z)$
- $C_Z = E_{Kz}(ID_B \,||\, M)$

34) What is an Onion Routing anonymity system?

E' il design principale per i Low-latency anonymity systems, più veloci rispetto ai mix-based systems ma anche più vulnerabili ad attacchi che comportano l'analisi del traffico. Consiste in un insieme R = {R1,R2, . . . ,Rn} di server chiamati Onion Router (OR) che inoltrano il traffico per i client, il primo passo è la creazione di una connessione anonima formata da tunnel criptati multipli attraverso la rete costituita da una sequenza di k OR; ogni OR memorizza chiavi di crittografia, ID di connessione, ID dell'OR precedente e successivo.

L'iniziatore seleziona innanzitutto una sequenza ordinata di k OR nella rete da utilizzare come percorso del circuito, genera due chiavi simmetriche per ogni OR lungo il percorso: una chiave di inoltro utilizzata da KFi per crittografare i dati inviati dall'iniziatore verso il risponditore e un KBi chiave a ritroso applicato ai dati dal risponditore all'iniziatore. L'iniziatore invia l'onion, insieme a un ID di connessione scelto CI_x, al primo OR nel percorso, Rx che rimuove dall'onion lo strato più esterno di crittografia usando la sua chiave privata e apprende le chiavi simmetriche KFx e KBx generate dall'iniziatore, nonché il server successivo nel percorso. Rx quindi padda il payload crittografato rimanente con byte casuali, in modo che l'onion mantenga una lunghezza costante e invia il risultato, insieme a un nuovo ID di connessione CI_y, a Ry.

35) What is the different between a sniffing attack and a Man-In-The-Middle attack?

Un attacco di sniffing si verifica quando un aggressore intercetta il traffico di rete tra due dispositivi. L'aggressore può utilizzare network sniffer per catturare e analizzare i pacchetti di dati che transitano sulla rete.

D'altra parte, un attacco di Man-In-The-Middle si verifica quando un aggressore si inserisce tra due parti che stanno comunicando, impersonando entrambe le parti. L'aggressore riesce a intercettare, manipolare o interrompere la comunicazione tra le due parti senza che loro se ne accorgano. Questo tipo di attacco può consentire all'aggressore di modificare o iniettare dati nella comunicazione oltre che a poterli intercettare ed analizzare.

36) Describe what a spoofing attack is.

È un attacco attivo, in cui un'entità (utente malintenzionato) si maschera con successo da un'altra falsificando l'origine dei dati (spoofing). Esistono spoofing su diversi layer di rete:

- Layer 2 (Link): ARP spoofing, in cui l'aggressore manda finti ARP per associare l'indirizzo MAC dell'utente malintenzionato all'indirizzo IP di un altro host, facendo sì che qualsiasi traffico destinato a tale indirizzo IP venga invece inviato all'utente malintenzionato.
  Il MAC spoofing è un altro tipo di attacco di spoofing: inonda la LAN di pacchetti con un indirizzo MAC di origine falsificato, questo processo "ruba" la porta dello switch dell'host falsificato
- Layer 3 (IP): IP Spoofing, in cui l'aggressore falsifica l'indirizzo IP di origine di un pacchetto di rete, facendo sembrare che provenga da una fonte affidabile o attendibile.
- Layer 4 (Transport): TCP spoofing, in cui un aggressore cerca di impersonare o falsificare l'indirizzo IP sorgente di un pacchetto TCP al fine di ingannare i dispositivi di origine e destinazione coinvolti nella comunicazione (source routing e blind attack).
- Layer 7 (Application layer): Email Spoofing, Webpage Spoofing etc.

37) Describe what a DoD attack is.

Attacco per rendere una macchina o una risorsa di rete non disponibile per gli utenti previsti. La tecnica più comune è il flooding: consuma le risorse delle vittime (banda della rete e/o capacità di elaborazione). Può essere eseguito da un singolo nodo o da due o più nodi. Un esempio è l'attacco SYN flood che sfrutta la debolezza nel protocollo TCP per saturare un sistema con richieste di connessione incomplete. Un smurf attack è un DoS flooding che sfrutta la possibilità di inviare pacchetti a tutti gli host di computer su una

particolare rete tramite l'indirizzo di trasmissione della rete: l'attaccante invia un gran numero di pacchetti con l'indirizzo di origine falsificato per sembrare l'indirizzo della vittima, di conseguenza la rete funge quindi da amplificatore dello smurf tentando di rispondere alle richieste.

38) Differences between a network scanner and a vulnerability scanner.

Un vulnerability scanner è uno strumento che viene utilizzato specificamente per individuare e valutare le vulnerabilità e le debolezze presenti nei sistemi informatici. Utilizza un database di conoscenze sulle vulnerabilità conosciute e confronta queste informazioni con la configurazione e lo stato dei dispositivi scansionati. Un network scanner è uno strumento che viene utilizzato per una panoramica della rete e identificare i dispositivi connessi. Un network scanner non si concentra specificamente sull'individuazione di vulnerabilità o debolezze nei sistemi.

39) Describe what a buffer overflow attack is.

Il buffer overflow si verifica quando un programma accetta input da un utente o da una sorgente esterna e non verifica correttamente la dimensione dell'input prima di copiarlo in un buffer di memoria. Se l'input supera la dimensione massima prevista per il buffer, il resto dell'input può sovrascrivere l'area di memoria adiacente, inclusi dati importanti come indirizzi di memoria, puntatori o registri di istruzioni.
Gli attaccanti sfruttano questa vulnerabilità inviando input appositamente progettati per sovraccaricare il buffer e sovrascrivere l'area di memoria adiacente con codice dannoso. Una volta che il codice dannoso viene eseguito, l'attaccante può ottenere il controllo del programma o del sistema,

40) What is a SQL injection attack?

L'obiettivo principale di un attacco di SQL injection è ottenere un accesso non autorizzato ai dati del database o compromettere il funzionamento dell'applicazione. Gli aggressori sfruttano l'iniezione SQL inserendo deliberatamente stringhe di caratteri dannose o comandi SQL nelle caselle di input dell'applicazione web, come i campi di login o di ricerca. Se l'applicazione non gestisce correttamente l'input dell'utente, l'iniezione SQL può consentire all'attaccante di manipolare le query eseguite dal database.

41) What is a packet filtering firewall?

È uno strumento di basso livello che non consente ai singoli pacchetti di passare attraverso il firewall a meno che non corrispondano al set di regole stabilito, è possibile filtrare il traffico in base a molti attributi di pacchetto. L’insieme delle regole è definito da un amministratore del firewall oppure è predefinito. Le regole di filtraggio sono elencate in elenchi o tabelle appropriati, denominati Access control lists.

42) What is an Intrusion Detection System?

Gli IDS sono sistemi software o hardware che automatizzano il processo di monitoraggio degli eventi che si verificano in un sistema informatico o in una rete, analizzandoli alla ricerca di segni di intrusioni. Cercano di scoprire tentativi di compromissione o aggiramento dei meccanismi di sicurezza di un computer o di una rete. Gli IDS elaborano un flusso di eventi E1, E2, E3, ... e gli stati di sistema passati S1, S2, S3, ..., e decidono se un nuovo evento E4 in S4 è la prova finale che si sta verificando un'intrusione: analizzano la manifestazione di un attacco, non il risultato dell'attacco.